SunBeam Institute of Information Technology

Java Assignment 6

Assignment based on Inheritance and polymorphism

Q1) Solve assignment to apply inheritance n polymorphism

a) Can you arrange Fruit,Apple,Orange,Mango in inheritance hierarchy ? Use encapsulation.

b) Properties (instance variables) : color : String , weight : double , name:String, isFresh : boolean

c) Add suitable constructors.

d) Override toString correctly to return state of all fruits (including : name ,color , weight)

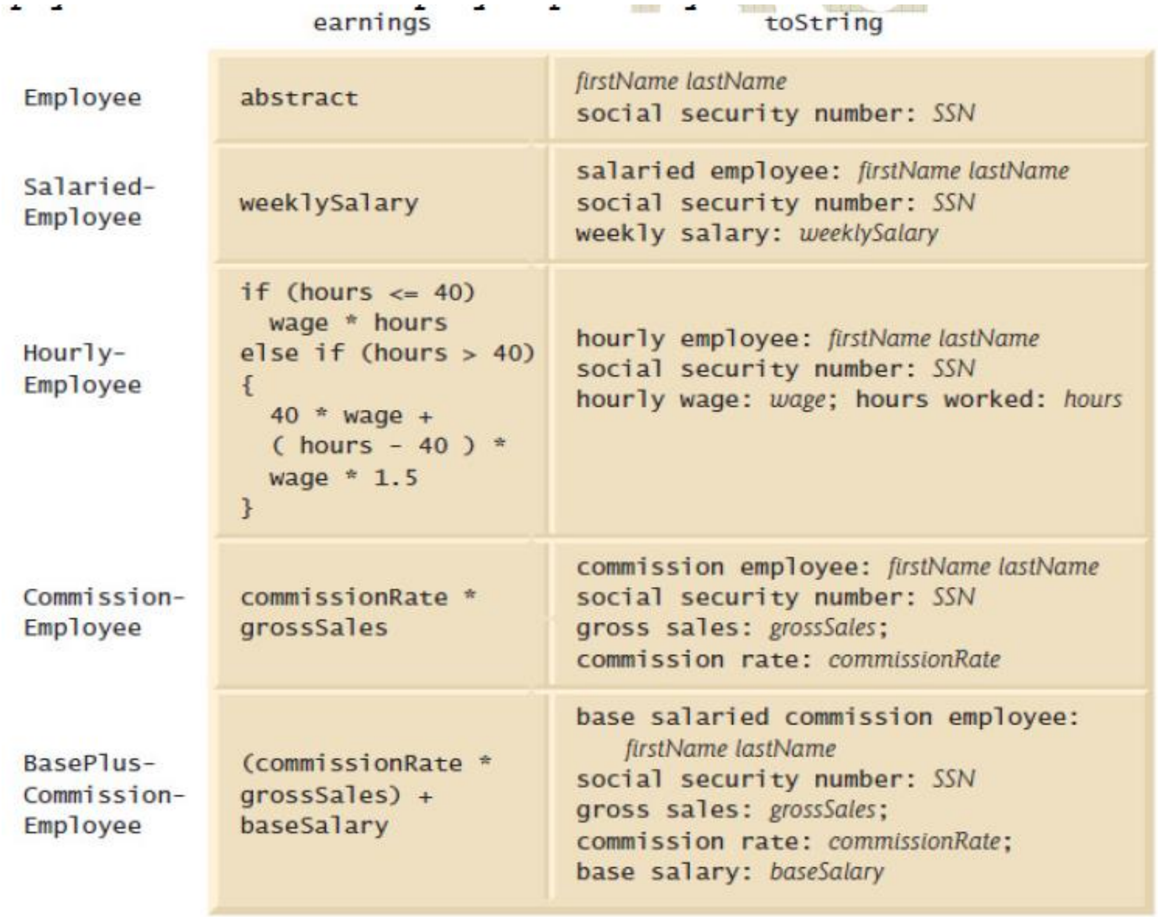
e) Add a taste() method : public String taste()

For Fruit : it should return "no specific taste"
Apple : should return "sweet n sour"
Mango : should return "sweet"
Orange : should return "sour"

f) Add all of above classes under the package "com.app.fruits"
g) Create java application FruitBasket , with main method , as a tester
h) Prompt user for the basket size n create suitable data structure and give options
1. Add Mango
case 1 : boundary checking
basket[counter++]=new Mango(nm,weight,color);
break;
2. Add Orange
3. Add Apple
NOTE : You will be adding a fresh fruit in the basket , in all of above options.
4. Display names of all fruits in the basket.
eg : for-each --- null checking --getName()
5. Display name,color,weight , taste of all fresh fruits , in the basket.
eg : for-each , null checking --toString , taste, isFresh : getter
6. Display tastes of all stale(not fresh) fruits in the basket.
7. Mark a fruit as stale

i/p : index
eg : setter : isFresh : false
o/p : error message (in case of invalid index) or mark it stale
8. Mark all sour fruits stale (optional)
eg : for-each , taste --equals(String)
10. Exit

Q2) A company pays its employees on a weekly basis. The employees are of four types:
Salaried employees are paid a fixed weekly salary regardless of the number of hours worked, hourly employees are paid by the hour and receive overtime pay (i.e., 1.5 times their hourly salary rate) for all hours worked in excess of 40 hours, commission employees are paid a percentage of their sales and base-salaried commission employees receive a base salary plus a percentage of their sales. For the current pay period, the company has decided to reward salaried-commission employees by adding 10% to their base salaries.
The company wants to write an application that performs its payroll calculations polymorphically.



	earnings	toString
Employee	abstract	*firstName lastName* social security number: *SSN*
Salaried-Employee	weeklySalary	salaried employee: *firstName lastName* social security number: *SSN* weekly salary: *weeklySalary*

Hourly-Employee	if (hours <= 40)
	  wage * hours
	else if (hours > 40)
	{
	  40 * wage +
	  (hours - 40) *
	  wage * 1.5
	}
		hourly employee: *firstName lastName*
		social security number: *SSN*
		hourly wage: *wage*; hours worked: *hours*

Commission-Employee	commissionRate * grossSales	commission employee: *firstName lastName* social security number: *SSN* gross sales: *grossSales*; commission rate: *commissionRate*
BasePlus-Commission-Employee	(commissionRate * grossSales) + baseSalary	base salaried commission employee: *firstName lastName* social security number: *SSN* gross sales: *grossSales*; commission rate: *commissionRate*; base salary: *baseSalary*

Optional Assignment:

Create Emp based organization structure with : Emp , Mgr , Worker

All of above classes must be in package--com.app.org

a) Emp state--- id(int), name, deptId(string) , basic(double)

Emp id should NOT be accepted from user. It should be auto generated , in auto increment manner

Hint : add static data member : idGenerator

Increment it and assign it to emp id in emp class constructor.

Get emp details -- using overriding form of toString

b) Mgr state ---id,name,basic,deptId , performanceBonus

Behaviour ----1. get mgr details : via toString

compute net salary (formula: basic+perfmonceBonus)

get performance bonus.

c) Worker state --id,name,basic,deptId,hoursWorked,hourlyRate

Behaviour---

get worker details -- via toString

compute net salary (formula: = basic+(hoursWorked*hourlyRate)

get hrlyRate of the worker -- add a new method to return hourly rate of a worker.(getter)

Can you organize these classes in inheritance hierarchy?

d) Write TestOrganization in "tester" package.

Create suitable array to store organization details. Provide following options. Run the application till "exit"

1. Hire Manager

2. Hire Worker

3. Display information of all employees , including net salary using single for-each loop.

Display from the same for-each loop, performance bonus if it's a manager or if it's a worker , display hourly rate of the worker .

4. Update basic salary

i/p : emp id & increment in basic component of salary

o/p : In case of invalid emp id , give suitable error message. Otherwise , increment the salary.

5. Exit